Introducing Chemical Computation Early in the Undergraduate Chemistry Curriculum

Katherine A. Kantardjieff,* Steven A. Hardinger,† and W. Van Willis
Department of Chemistry and Biochemistry, California State University, Fullerton, Fullerton, CA 92834;
**kkantardjieff@fullerton.edu*

Instruction in the physical sciences is commonly carried out using the "problem-solving" methodology. Students are presented with classical examples that illustrate the applicability of the subject, and they are expected to master the mechanics or algorithms of these examples and be able to recognize new or unfamiliar situations where these skills can be applied. This works for some but not all students (*1*, *2*), and it is essentially a passive learning mode (*3*). Moreover, such an instructional process is rather different from the actual practice of the disciplines. Practitioners of science spend a great deal of time dealing with data, analyzing it, searching for connections within it, and representing it in different ways to better understand it. They also spend time examining physical and symbolic models to understand their applicability and limitations and to extract further information.

The computational facility in the Department of Chemistry and Biochemistry, established jointly with the Department of Physics, serves as an electronic classroom where students explore models and data in the chemistry and physics curricula. It is equipped with 10 Silicon Graphics workstations and Power Challenge server, networked together and to the campus Internet backbone. Thus, our students and faculty are linked to resources and courses with common themes but traditionally segregated. The goals of this facility are:

> To build on the basic science that is shared by chemistry and physics. This cooperative endeavor will lead to better correlation of courses within each discipline as well as between the disciplines.
>
> To supplement traditionally passive lecture time with exploration of data and models, which are organized to illustrate important concepts and principles that cannot be treated easily or well by the problem-solving methodology.
>
> To systematically introduce students to exploration activities that utilize various computer applications. Students first explore pre-prepared files during class under direct guidance of instructors. With experience, they apply what they have learned to more sophisticated problems and exploration, eventually moving to free-form assignments that allow them to explore specific content in detail and to test limitations of the theory and of the applications.
>
> To show students (and instructors) that chemical and physical concepts can be learned using the computer as an assistant in the same way that our predecessors used graph paper, slide rules, and balls and sticks. The scientific method hasn't changed. The new (software) tools are arguably better.

Achieving these goals required that our students be introduced to computers early in the chemistry curriculum. Prior to 1993, we required students to complete a course in elementary computer programming. We found, however, that they were not transferring the skills learned to the chemistry curriculum. We have worked out a way to put computers as tools into the hands of our students.

Introduction to Chemical Computation

In the spring of 1993, we initiated a one-semester, 2-unit sophomore-level course, CHEM210, "Introduction to Chemical Computation", that has since replaced the computer programming course requirement in our major. This course gives students experience in the use of modern software to solve real chemical problems. The prerequisite is one year of general chemistry, and it is assumed that students have some basic knowledge of word processing. The content of the course has evolved with the acquisition of more powerful computer systems, the establishment of several electronic classrooms, and the involvement of several faculty, but the focus is on *chemical applications*. In CHEM210, students learn how to use various software packages as they learn how to develop a logical sequence of steps towards solving a chemical problem or investigating a molecular system. Students learn the fundamentals of (i) operating systems (DOS, MAC-OS, UNIX); (ii) data transfer (Telnet, FTP, email); (iii) navigating the Internet (Netscape); (iv) html (Web Wizard, asWedit); (v) software for molecular modeling and chemical computation (Spartan, RasMol); (vi) mathematical manipulation (Excel, Mathematica); (vii) presentation development (PowerPoint); and (viii) several electronic library resources. Students are not required to purchase software or computer time other than a campus-wide computer lab fee card. A typical syllabus, problem sets, and index of student home pages may be found at the CHEM210 Web site (*4*). Traditional passive lecture format is not used. Students learn at the computer terminal during interactive sessions. The instructors serve as facilitators, guiding students through the software tools and, at the same time, discussing problem-solving strategies and interesting results. Most of the assignments are exchanged electronically. Students post graphical results and the results of their database searches on a personal home page. In addition to regular assignments for each application, the students utilize the tools they have mastered in CHEM210 to complete an independent investigation of a chemical problem. The results of their investigation are published on their home page.

Presentation Software

The PowerPoint software package (Microsoft Corp.) is taught early in the class because it is quickly mastered and provides students with an opportunity to review and master many simple skills needed for general computer use. They are introduced to the basic elements of visual design, content selection and creation, and importing graphic images. They are then asked to create a four- to six-slide presentation

†Present address: Department of Chemistry and Biochemistry, UCLA, 405 Hilgard Ave. Los Angeles, CA 90095.

describing the osmotic pressure of a 1:1 electrolyte, and later to design a graphic to include on their home page.

Electronic Library Resources

Students are introduced to the chemical literature online in two sessions. One session reviews scientific information freely available through the Internet at several sites such as the ChemCenter page (*5*) at the American Chemical Society's Web site. The other session covers resources available through our library's computer system and Internet connection, which has special access privileges to several scientific subscription databases. These databases include First Search (OCLC Corp.) (*6*), MELVYL (University of California) (*7*), Science Citation Index (Institute for Scientific Information), CARL UnCover (Carl Corp.)(*8*), and Chemical Abstracts Service Online (Chemical Abstracts Service)(*9*). Some databases such as MELVYL and First Search have command line user interfaces and require searches to be designed in a syntax peculiar to the database. Although quite useful and efficient, these tend to be less popular than those relying on standard English terms or menu-driven search options. Access to certain databases such as CAS Online is carefully controlled to limit costs. A typical student assignment would be to find, retrieve, and cite several recent research articles on a molecule of their choice, such as one they have selected to show graphically on their home page.

Molecular Modeling

Molecular modeling and computation are integral to the practice of modern chemistry. The level of presentation in CHEM210 is aimed at students who are currently taking or have recently finished at least one semester of organic chemistry, but who have not yet taken physical chemistry. We begin with discussion of why chemists use models. This leads nicely into an interactive session on geometry optimization, molecular mechanics and force fields. Molecular mechanics calculations can give a structure, some spectroscopic data, a dipole moment, and the heat of formation, from which the strain energy can be evaluated. The calculations, however, owing to their model nature, are best regarded as good solutions to a set of simultaneous equations specific to one or two physical properties. This leads to calculations based on electronic interactions and discussion of ab initio methods. It is possible to obtain chemical accuracy using ab initio calculations, but the cost in computer time is enormous, and only small systems can be treated this accurately. We then move on to discussion of semiempirical methods. Students work through the tutorial provided with the Spartan computational package (*10*, *11*), an exercise that occupies most of them for from one to four hours before they feel they have sufficient mastery of the software to tackle the problem sets.

Problem sets, which we revise each semester, emphasize fundamental chemical concepts, demonstrate various features of Spartan and molecular modeling in general, and introduce some new chemistry. The first problem set contains fairly detailed instructions on building structures, setting up calculations, and recording results. Students evaluate the quality of computational results vs CPU time for the calculation, use heats of formation to decide on best molecular geometry for CH_4 (familiar to students) and XeF_2 (a much less obvious case), explore resonance in SO_4^{2-}, and do some peptide modeling.

The second problem set contains significantly less detail, which facilitates the students' development as independent problem-solvers. In one activity, Bredt's rule is explored with Spartan. In another, students search the Brookhaven Protein Data Bank (*12*) via Netscape and select and download a protein structure, which they then manipulate with RasMol (*13*). A GIF file of the final rendering of the structure, along with a brief explanation, is emailed to the instructor.

Student performance on these two modeling assignments is evaluated on the basis of chemical accuracy and an understanding of the basic workings of the software. Special consideration is given to students who show exceptional understanding or insight, and grading is done with consideration of the student's chemistry background.

Mathematical Manipulation

The aim of this portion of the course is not to teach the students computer programming per se. Rather, students learn how to use spreadsheets and symbolic algebra software while learning how to develop a logical sequence of steps toward solving a chemical problem or investigating a molecular system. Students meet for two interactive sessions using Excel (Microsoft Corp.). During the first session, they are introduced to data entry, cell formatting, and basic mathematical calculations, including built-in functions that are commonly employed in the sciences. The second session concentrates on more sophisticated techniques, such as graphing, linear regression, and construction of a database that can be linked and easily searched. Students are then asked to solve increasingly more sophisticated problems.

The students begin by constructing a spreadsheet that will monitor the chemical stockroom at CSU Fullerton. This exercise helps them learn about formatting a usable spreadsheet, data entry, sorting, and achieving simple calculations. Next, they make array calculations involving the ideal gas law. This helps them to understand the construction of repetitive calculations (the analog of a "do loop") and manipulation of arrays. Later they use these data to make 2- and 3-dimensional graphs. Other problems involve use of built-in functions such as natural logs and trigonometric functions. Linear regression is learned using a Beer's law plot of spectroscopic absorption data and kinetic data. The students are then asked to work in pairs, each partner constructing a database of the periodic table containing information about the elements that is different from the other partner's. They then demonstrate that these spreadsheets can be successfully linked and searched, and that the results of the search can be output as a separate report. Finally, the limits of the mathematical capabilities of the spreadsheet are demonstrated with an analysis of a particular chemical equilibrium. This involves devising an algorithm that computes concentrations and asks questions about the chemical system using various scenarios.

In the Mathematica portion of the course, which meets for two interactive sessions, students are introduced to very basic Mathematica scripting (*14*). They learn how to construct a simple instruction, a repetitive calculation, and 2- and 3-dimensional graphs, and to manipulate arrays. The ideal gas law problems from the Excel problem set are revisited here, as is the analysis of the chemical equilibrium. The students are also asked to construct a titration curve for a weak acid.

Assessment

At the end of each of the last four semesters, we have asked the students (68 in all) to anonymously fill out a questionnaire about their general computer usage and their experiences in the class (Tables 1 and 2). These surveys help instructors to make appropriate modifications to exercises, problems, and projects, removing those that the students find confusing or that the instructors find to be of little value and introducing new ones that reinforce student learning and challenge them to explore, to function as practitioners of science.

Table 1. CHEM210 Survey: Application Preference

Semester	Application[a]								
	Excel	First Search	Mathematica	MELVYL	Netscape	PowerPoint	Rasmol	Spartan	html
Fall 1995	2.9	–	3.7	4.0	1.0	1.6	–	1.0	–
Spring 1996	2.6	1.7	2.6	–	1.8	3.5	1.5	1.9	–
Fall 1996	3.0	2.4	2.6	–	1.6	2.4	2.0	2.3	1.6
Spring 1997	1.7	1.5	2.0	–	1.0	1.7	1.3	1.9	1.9

[a] Ratings are on a scale of 1 (liked very much) to 5 (disliked very much). A – indicates that the application was not used during that semester.

Table 2. CHEM210 Survey: Computer Use, Computer Literacy, and Course Format

Item	Survey Result
Time spent on assignments outside of class	1–10 h/wk (5 av)
Time using computers outside of class in other courses	0–20 h/wk (10 av)
Self-described computer literacy before taking course[a]	low 56%; medium 34%
Students having modems at home for remote access	33%
Previous knowledge of applications for:	
word processing	77%
graphing	46%
spreadsheets	36%
databases	15%
communications	15%
Preference for more than one instructor	55%

[a] On a scale of none/low/medium/high/expert.

Student comments were generally favorable for those programs that were the most intuitive for them and had the best online help and tutorials. Students indicated that they would likely use all of the skills they had learned in CHEM210 in most of their other chemistry courses for a variety of purposes, including research, written and oral presentations, and molecular visualization. Students' frustrations with Mathematica during the first semester and PowerPoint the second semester were due, in part, to software problems and file incompatibilities between computer platforms. We believe that Mathematica and Excel were ranked low initially owing in part to the students' lack of exposure to mathematical models for experimental phenomena in previous courses and lack of analytical skills in problem-solving. Without question, these two software packages are less intuitive and more complicated to use, and they offer minimal online support for the user. A more telling point is that students have not been taught to view a mathematical model such as the ideal gas equation as a tool for modeling experimental data and making quantitative predictions.

Computing Facilities

The electronic classroom provides faculty and students access to 10 Silicon Graphics 100-MHz Indigo R4400 PCs, each with 32 MB memory and 0.5 GB system disk and networked to a Power Challenge server. The server is configured with two 75-MHz TFP CPUs, 256 MB memory, and 12 GB of system disk storage. The computer array is completed by an Indigo2 150-MHz R8000 Extreme with 64 MB memory and 2- and 4-GB disks, located in the X-ray diffraction laboratory of CMolS. The entire array is networked by Ethernet TCP/IP. The facility is maintained by a full-time department systems administrator. During the academic year, the facility is open 35 hours per week for student use, except when booked for formal classroom activities. Students are issued accounts when they enroll in a specific course. Six student assistants help students with software applications and projects, answering technical questions and maintaining security. Additional school-wide facilities, equipped with several dozen Power Macs and WindowsNT platforms, are also used by students.

Available software includes molecular modeling and refinement modules and databases (CSD, NIST), sequence and homology searching modules, and solids and polymer building modules from Biosym/Molecular Simulations, Spartan (Wavefunction, Inc.), Hyperchem (Hypercube, Inc.), Mathematica (Wolfram, Inc.), and Gaussian94 (Gaussian, Inc).

Discussion

Several years of pilot studies preceded the development of CHEM210 as described in this article. Other college-level courses are designed to teach more sophisticated computer programming and system operations. The activities developed in CHEM210, while aiding the students to learn how to use computational tools in a hierarchical fashion, are about *chemistry*. For both students and faculty to use computers as tools for teaching and learning in this way, the transition must be relatively easy to be effective, so that students can focus on the content of the activities in which they are actively engaged, faculty can make assignments and monitor progress of students, and neither is distracted by having to learn the idiosyncratic details of the hardware.

A common stumbling block in the mathematical manipulation portion of the course is not mastery of the software, but a lack of a firm foundation in the basic chemistry. Students, particularly those whose approach to problem-solving has been to memorize each problem rather than to establish a logical mechanism for solving a type of problem, often have difficulty developing an actual algorithm for determining an answer to the question being posed. They make a distinction between a "computer" class and a "chemistry" class, rather than seeing the computer as a tool to do their chemistry. Once they overcome this hurdle (which usually means some review of first-year chemistry notes), they become enthusiastic participants. We have found also, as student surveys suggest, that students use the software in other courses to solve problems and analyze data. Use of spreadsheets, in particular,

has become commonplace in many of our laboratory courses, and spreadsheets serve as interfaces to instruments in many of our undergraduate research laboratories.

Our students have responded to the electronic classroom and CHEM210 with great enthusiasm. With modest guidance and supervision, they learn to use applications and, in fact, quickly become reasonably sophisticated users. They devour manuals and online help with greater alacrity than do faculty, they learn by teaching others, and they desire to do more. Student assistants, who are described in the computer facility schedule by their area of expertise, are an excellent source of support. These assistants are often selected from among students who have completed CHEM210.

Introduction of chemical computation early in the curriculum has also had a positive pedagogical impact. Students benefit from instruction by faculty whose expertise is with particular computer applications in chemistry, and faculty benefit from creative discussion as exploration activities are developed and evaluated. We continue to work to better correlate courses within chemistry, so that we coherently expose our students to more sophisticated problems and exploration. Using computational tools, they will begin to see the interconnections among topics in chemistry and learn how these topics can be viewed and described from different perspectives.

Acknowledgments

Initial funding for the active-learning computer facility was provided by NSF-ILI-DUE. Additional funding for instrumentation and technical support was provided by NSF-ARI-BIR, NeXstar, Inc., and CSUPERB (California State Program for Education and Research in Biotechnology). Funding for some software development was provided by Research Corporation Partners in Science.

We would like to thank our colleagues for their creative contributions to curricular development and enthusiastic discussions: Patrick Wegner (Cal State Fullerton, Chemistry); James Feagin, Heidi Fearn (Cal State Fullerton, Physics); Russell Cramm (Downey High School, graduate student, Cal State Fullerton, Chemistry). Special thanks go to those who have maintained hardware and software and assisted students: Brad Van Mourik, Herb Axelrod, Brian Schick, Guy Crundwell, Louis Larres, Robert Igarashi, Richard McClellan, Michael McCoy, Michael Goddard, Richard Castillo, Mark Kottke, Nimesh Udani, Jerry Bhatia, Mandana Moghadam.

Literature Cited

1. Tobias, S. *They're Not Dumb, They're Different; Stalking the Second Tier*; Research Corporation: Tucson, AZ, 1991.
2. Hewitt, N. H.; Seymour, E. Factors Contributing to High Attrition Rates Among Science, Mathematics and Engineering Undergraduate Majors, Alfred P. Sloan Foundation: New York, 1991.
3. DeBoer, G. *A History of Ideas in Science Education*; Teachers College Press: New York, 1991.
4. URL: *http://zeppo.fullerton.edu:8080/~kkant/210syl.html* (accessed Mar 1999).
5. ChemCenter; *http://www.chemcenter.org* (accessed Feb 1999).
6. OCLC; *http://www.oclc.org* (accessed Feb 1999).
7. MELVYL System Home Page; *http://www.dla.ucop.edu* (accessed Feb 1999).
8. Databases (Colorado Alliance of Research Libraries); *http://horus.coalliance.org/database.html* (accessed Feb 1999).
9. CAS; *http://www.cas.org* (accessed Feb 1999).
10. Hehre, W. J.; Huang, W. W. *Chemistry with Computation*; Wavefunction: Irvine, CA, 1995.
11. Hehre, W. J.; Burke, L. D.; Shusterman, A. J.; Pietro, W. J. Experiments in Computational Organic Chemistry; Wavefunction: Irvine, CA, 1993.
12. URL: *http://pdb.pdb.bnl.gov* (accessed Feb 1999).
13. Sayle, R. RasMol v2.5, Molecular Visualization Program; Glaxo-Wellcome: Greenford, Middlesex, UK, 1994.
14. Wolfram, S. *Mathematica, A System for Doing Mathematics by Computer*; Addison-Wesley: Reading, MA, 1992.